July 2018

Issue 22

THE SPECTRUM SHOW

Magazine



MAKING CODE ZERO

FULL DEV DIARY

RD DIGITAL TRACER

TRACE AND DRAW

PLUS

THE ZX PRINTER

Smell the burn..

FLASHBACK 87

GAME REVIEWS

FEATURES

HARDWARE

Includes material not in the video show!

MIND YOUR LANGUAGE

Continued programming feature

CONTENTS





32. DIGITAL TRACER
Tracing images.



22. MIND YOUR LANGUAGE
Spectrum Forth.

FEATURES

4 News from 1987
Find out what was happening back in 1987.

16 Making Code Zero
Full development Diary.

24 Mind Your Language
More languages from George.

32 RD Digital Tracer
Tracing things on your Spectrum.

34 Revival 2018
Report from the recent event.

36 Grumpy Ogre
Retro adventuring.

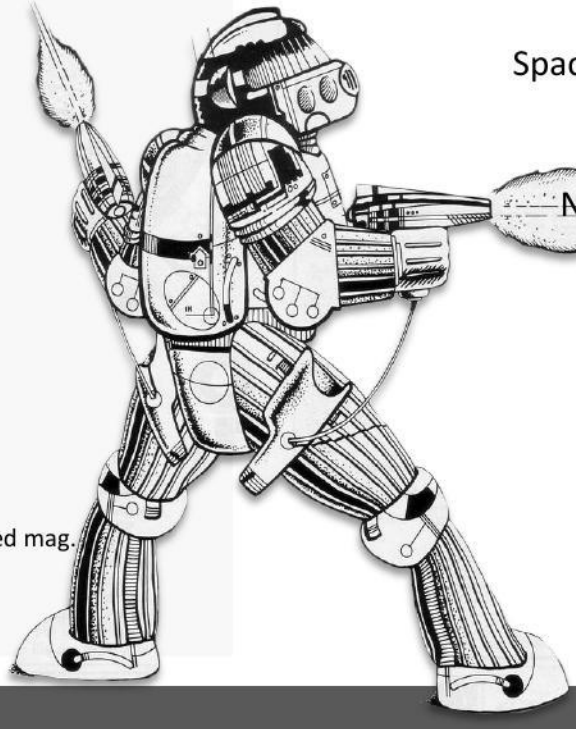
40 ZX Printer
The tiny Sinclair printer.

42 16/48 Magazine
A new series looking at this tape based mag.

And more....

GAME REVIEWS

Pitfall 2 6
Panama Joe 8
Out Run 9
Vallation 10
Aquarius 11
Space Invaders Emulator 12
Gobbleman 13
Nixy The Glade Sprite 18
Planet Of Death 20
On The Run 22
Kokotoni Wilf 23
And more...



Cover art by Fandi Winata
(https://www.artstation.comfandiw)

EDITORIAL



Welcome to issue 22 and thank you for taking the time to download and read it.

The previous issue had a less than smooth release. Initial versions had the wrong text for a game review, which obviously didn't make sense and subsequent versions had page numbering issues and text missing from the Mind Your Language section. The penultimate version, for some reason, had a large image of the Omni missing after conversion to PDF. Eventually though, the correct version was uploaded. If your version has incorrect page numbers or no half page image in the Omni review, please grab the new version.

New Game

My new game is now available to download free of charge. It is based on yet another of my old BASIC games from the 80's and will have a full 'making of' feature in this magazine soon. It was great to see the game I originally envisaged all those years ago, finally looking like the image I had in my head.

ZXDB Additions

I was approached by Peter from the SpectrumComputing website (details on page 7), and asked if I had a list of all the games I had reviewed on The Spectrum Show. I didn't, but was more than happy to catalogue them. Now, if a game has been reviewed on the show, a link will take you from the page on the website to the exact time in the video of the review. This is a great addition to the database, and I hope to keep adding to this as each new show is released.

Revival 2018 was held at Walsall again, and this year it was double the size. It was fantastic to catch up with a few friends, browse the trader's stalls, grab some games on old consoles and of course the many arcade games on offer.

The whole atmosphere is different from the other events I visit, a more homely, welcoming retro feel. Pictures can be found on page 34.

You never know who you'll bump into at these events as there are many people from the retro scene including podcasters, youtubers and legends from the scene like Jim Bagley and Simon Butler. I bumped into Octavious Kitten too, which was a nice surprise!

I also bumped into Peter from ZXrenew and after some discussion I came away with a few items to play with. I was already planning to buy these, but with Peter on hand it was an ideal opportunity to grab a few different colours to see how they worked together (see the advert for ZXrenew on page 21). After a few hours I had a beautiful white Speccy with a silver faceplate and LED membrane.



Anyone following my twitter feed will know that I recently blew three of my Spectrums up! It wasn't spectacular, just a corrupted display, but it was a terrible feeling as one after the other displayed the same fault.

Now I must point out that this didn't happen due to the age of the machine, it was due to something I have never come across before, a faulty ROM cartridge. At least that is my only conclusion after reviewing the evidence.



All three machines worked fine. All three machines had previously worked with the same Interface 2 on many occasions. All three machines worked fine with all my other ROM games. All three machines died after inserting my newest game, Planetoids. Planetoids had corrupt graphics on all three machines. If anyone else has any other explanation, please let me know. I am happy to loan the offending cart to anyone wishing to investigate further but take no responsibility for damage it may cause!

Fancy writing a game review or special feature?

I am always looking for new content and all contributions are welcome.

Z88 PROBLEMS



The Ministry of Defence is said to be contacting many companies with a view to copyright infringement of its patented LCD screen technology.

The patent is for LCD screens to produce a clear image close to that of a conventional CRT, and rumours are that several companies, including Sinclair are using it without permission.

Although the patent is in place in the UK, USA and Canada, it has yet to reach Japan, where most of the components are manufactured.

How this move will affect Sinclair's new Z88 computer has yet to be defined, maybe a price increase, maybe even withdrawal, but Clive Sinclair says he has not yet been contacted by the MOD, so cannot comment on the situation.

Sinclair have stated that the Z88 will be delayed, and will not now be in full production for another six to eight weeks. If this due to the screen concern, we'll never know.

The company are also coming under fire for advertising the product and taking orders without actually stating when the delivery date will be.

PRISM GET ADDICTIVE

Addictive Games, the company run by Kevin Toms, has signed a deal with Prism Leisure that sees Prism take over not only the back catalogue, but also four future games including the newly released President.

As Addictive is absorbed into Prism, Kevin Toms will now be working under the new name of KJT Design Limited.

GREMLIN INVADE CANADA

Sheffield based software house Gremlin Graphics have signed a deal with Canadian development house Acme Animation that will give them the chance to get their titles into the US market.

Despite this potentially massive undertaking, Gremlin are still expanding in the UK, opening a third development studio in Derby, to go with the ones already set up in Sheffield and Birmingham.



IMPROVED CART



There is a new improved version of the humble Microdirve cartridge now available from Ablex.

The popularity has not faded since Sinclair sold the rights to Ablex six months ago, and they have now added a redesigned internal spring mechanism to improve reliability and performance.

OLD GAMES RETURN

Firebird have purchased the rights to most of Activision's back catalogue and will be releasing over 40 titles across the next 12 months. It's going to be a budget bonanza!

If you missed them the first time round you will be able to get your hands on titles such as Back To The Future, I of The Mask and Pitfall Two.

Don't be a keyboard hunchback!

Sit back with a compact, lightweight, hand held, fingertip operated joystick.

- *Nylon encased—Steel shafted joystick with ball and socket joint.*
- *Fast sprung return to centre.*
- *12 Months Guarantee.*
- *7 day Money back Guarantee.*

Settle down in comfort for that long battle against the high scores.

Large handles and rubber suckers are fine on a table top, but are very uncomfortable and tiring held in the hand and they don't stick carpets or sofas.
A delta 3S on the other hand will move with the flick of a finger.
Its choice of three fire buttons will suit a variety of grips in left or right handed operation. The rotary slide switches cannot be damaged by excessive pressure on the stick and the fire buttons have been extensively tested and will still be going after millions of missiles have been fired.
Like the Spectrum it is British with a sleek low profile style and matching colour scheme.
It will fit any interface that accepts a standard Atari type 9 way "D" plug and will give diagonal as well as vertical and horizontal movement.



DELTA 3s HANDSET £10.00
Price includes VAT and P&P.

Make the most of your joystick.

A programmable interface will work all your games programs.

Firmware type.
Easily programmed at any time.
Switch to A to program.
Switch to B to run.
Free! with firmware interface super "sound booster" which turns a "pathetic" peep into a "triumphant" tone.
Firmware programmable interface only £29.00.
Save £2! Delta 3s Joystick and firmware interface £37.00.

Software type.
Load the tape and select the joystick program that you want or make up another program and add it to the library.
Software interface only £28.00.
Save £2! Delta 3s Joystick and software interface £36.00.
Both interfaces have through connections for other add-ons.

Made in England

Voltmace *delta 3s*

Voltmace Limited
Park Drive
Baldock
Herts
SG7 6ET

PITFALL II

Activision 1984







Pitfall was an iconic game for the Atari 2600 and it was quickly followed by Pitfall II – The Lost Caverns.

Like the original, the game was ported to many systems including the Spectrum and had a huge reputation to live up to.

The game had large levels to explore, treasure to collect, people to rescue and jungle creatures to avoid and lots of platform action. This game also introduced save points, along with the most annoying aspect of the game for me, the death routine. But we'll come onto that in a moment.

The main character, Pitfall Harry is guided around the flip screen levels by walking, jumping, climbing or swimming and exploration is key. In many screens the treasure, in the case of the Spectrum this is gold bars, are hidden slightly off screen, so you have to guide Harry to the edges to make sure he has not missed anything.

The gameplay is nothing new, and wasn't when it was released, but it is quite impressive that Harry can do so many actions. The swimming is particularly impressive considering there are waves to navigate.

Walking left or right and jumping work well, as does climbing ladders. However under some circumstances you plummet to your death when trying to go down. This was just one of the many frustrations. When you approach the top of a ladder, you just drop down – there seems to be a lack of collision detection here, and this catches you out time and time again.

I later (much later) found out that to use the ladders properly, you have to be pressing down or up, when you move over them. This stops you falling to your death. This is a ridiculous control mechanism, and I can't believe it got past the testers!

If you die, which will be very often, Harry just floats slowly, very slowly, back to the last checkpoint. This is why





checkpoints are so important.

Many times I had navigated about ten screens, only to die and then have to sit and wait for Harry to float back ten screens! Very, very annoying. This kind of mechanic does not encourage the player to try again, it just makes them want to give up.

The graphics are adequate. They are not particularly detailed but are well animated and smooth. The backgrounds are bland, very bland. It's as though the authors are trying to replicate the Atari 2600 game on the Spectrum. But why?

The caverns are all represented by yellow platforms, and usually at the bottom you will find water. No variation (at least as long as I could bear to play it).

Sound is, argh! The music... I could not find a way to turn it off... and it drives you mad! It does change when you collect some treasure, but there should have been a way to turn it off.

Control is responsive, but there is a slight pause before you jump, which is a bit odd and can sometimes mean you die because the game has this built-in pause. Again, no real need for it unless they were deliberately trying to make the game harder.

Gameplay would be quite good if it wasn't for the awful death thing... it really does distract from the game and causes you to quickly lose interest.

Only play this if you are a fan... or a masochist..



PANAMA JOE

Parker / Sinclair Research 1984



Panama Joe is also known by the more common name of Montezuma's Revenge, and this was the name that the game used when it appeared on other systems, most famously the Atari 2600.

I have heard people say that the game was brilliant, but maybe that was just the Atari version, because I can't the brilliance in any aspect of it.

Maybe because it was an early game, or maybe because people fondly remember the Atari version, but for the Spectrum I expected more.

The game is, for want of a better phrase, a Jet Set Willy like multi-screen platform game with a few extras thrown in.

There are the usual ladders, poles and ropes, flaming pits, disappearing platforms and a menagerie of enemies scattered through the game, there are also screens with very little in them apart from a huge mass of bricks and a single ladder.

The aim is to reach Montezuma's treasure room, and you have to employ the uses of various items. There are things to pick up such as swords, that allow you to kill skulls, and various coloured keys that open the equivalent door. Some objects make the enemies harmless for a short period of time too.

Poles can only be slid down, ladders and ropes allow movement up and down. Some screens require jumps from ropes, and good timing to get past the enemy.

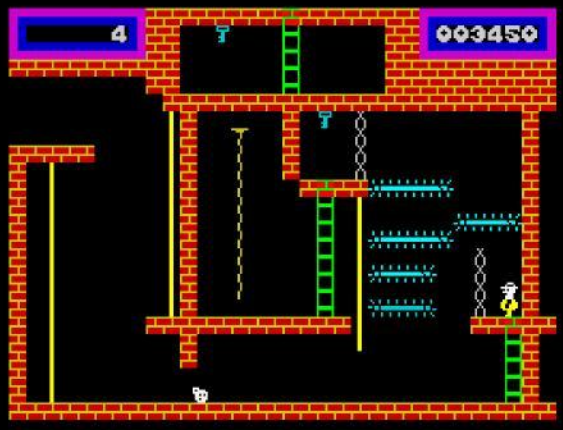
One room has a pole that always lands Joe on a skull so you have to make sure you a sword at this point.

There are different routes you can take to reach the treasure, but it is all down to trial and error. I used a key in the wrong place and was left with no other option than to restart again.

The graphics are very basic although Joe himself is nicely animated. The other graphics look like they have just been ported from the Atari!

Sound is non-existent when walking about, and you only get an effect when dying, opening doors or collecting something. The later plays a tune, but the game stops to do this, which is a bit tedious.

The map is quite small, so a determined player should eventually complete it. For me, though, I'll leave it alone thanks.



OUT RUN™

US Gold 1987



Arguably, one of the most notable and recognisable arcade racing games is Outrun. Fast paced action, great graphics and memorable soundtrack. Converting this to the 8-bit Spectrum would be a challenge.

In case you have never seen this, or any arcade style racer, the idea is to drive very fast and make it to the checkpoints before the time runs out. As you reach each checkpoint, the timer resets with any spare time added.

Each checkpoint has a choice, left or right. Left is the easier option, right is harder.

Is it me or does the game look like it's in slow motion. I don't think it's because I had just been playing the arcade version for comparison either, everything just looks so slow.

The graphics are nice and monochrome instead of full colour like the arcade, but they are quite detailed. At least you get plenty of time to look at them due to the speed of the game.

The track side objects are few and far between and the other vehicles seem to be just a splodge on screen.

The track does move up and down giving the impression of hills and valleys and of course it does sweep left and right, but it's like a slow motion car crash - which often happens in this game!

The music is nice on 128k machines, recreating the music of the arcade, but you have the option to play the real arcade music on the provided bonus tape. This was a nice touch.

Sadly I could not find a way to switch the music off so there is little in the way of engine sounds, just screeches when you go round corners or crash.

Because of the game speed the controls seems to be very slow in response, and can cause you to end up in the back of a lorry or arguing with a tree.

Once you slow your brain down to match the game speed not only does the game play a little better, but you are now also fully qualified to be a project manager!

A good attempt then by US Gold but somehow, after playing the arcade version, this one seems pedestrian.

I managed to get to the first checkpoint once in about 30 minutes playing and then proceeded to crash into almost everything.

Not a bad racing game, once you slow your brain functions down but certainly not the best.





Tardis Remakes 2016

This superb looking game has you trying to escape from an enemy complex without dying, and also destroying their ships and lasers where required, which is more or less everywhere.

After some great music, you are thrown into the game and quickly realise this is very much based on Cybernoid, and that's not a bad thing.

You control your ship, which has gravity, meaning you are always stabbing the up key to keep lined up with exits and corridors, this mechanic can make things difficult, but well designed screens limit the problem.

The first set of screens has things you can't destroy, so it's a matter of manoeuvring around them and this is a good introduction to both the game style and controls.

Later we get things to shoot, and things that shoot back and there are also missiles that launch upward. The game throws yet more hazards at you as you progress further.

The graphics are superb. Very well drawn, very colourful and very smooth.

Sound is used well, although things can be quiet for periods of time while you plan your route. This is always a good method; find a safe place on screen, watch the movement patterns and plan your route before heading off.



Control is crisp, meaning tight navigation of walls and aliens is, although not easy, at least the failure of yourself rather than the game.

The gameplay is, for me, a little tricky and sometimes unforgiving. In places you have to be pixel perfect to get through and this can lead to some frustration, but for good game players this will be a nice challenge. Practice and patience pays off though.



A great game then and definitely worth tracking down.

AQUARIUS

Bug Byte Software 1983



Aquarius is an early game as you can probably tell by the basic look of it. It sets you as a commander of a frogman team out to destroy death machines hidden underwater by evil governments around the world.

Before the game starts it displays the secret code required to destroy the machine, should you be lucky enough to ever reach it. This code comprises of three colours in sequence. Once you have memorised this, the game begins.

To get to these death machines you have to first navigate the dangers of the ocean. Things like jelly fish, sharks and mines. Contact with any of them will lead to your death.

You are armed with a spear gun and can use this to clear the way, even shooting the mines to clear your path.

You have a limited amount of oxygen too, but this can be replenished by collecting tanks thoughtfully left for you on the ocean floor. The inlay doesn't mention these and I found them by accident!

The screen slowly scrolls (character squares at a time) from right to left and you can move in all directions to dodge creatures and collect oxygen as required.

Jelly fish, sharks and mines appear at random and you have to either avoid or shoot them.

Jellyfish and mines just float about, squids move straight towards you and sharks swim across the screen, turn and head for you. This keeps going until you get to the caverns.



Once there, the game become a Scramble like game with things shooting vertically and electric force fields to deal with.

Once you get to the death machine, which is pretty unimpressive, you enter the secret code by shooting the correct coloured blocks.

Complete this and destroy the machine and it all starts again.

The graphics are very basic with limited animation and it certainly looks like an early game. That said, it is challenging to play.

Sound is used sparingly with bleeps when you fire, hit something or die.

If you can get over the bland graphics there is a decent game in there and not bad for a quick challenge.

THE SPACE INVADERS EMULATOR

Way back in episode 5 of the show, we looked at the Pacman emulator. A brilliant piece of code that ran the real Pacman arcade ROMS on your spectrum. A great achievement and at the time I said it must only be a matter of time before someone does the same for other, early arcade games. Well now they have.

The classic Space Invaders can now be fully emulated on your 128K Spectrum. Another stunning piece of work.

The emulator, released in 2017 by 40Crisis does not include the arcade ROMS – that would be illegal, but they are not difficult to find if you know how to use a search engine. Be careful though, some sites are not all joy and happiness. Make sure you have a good anti-virus program first!

Once you have the ROMS there is a simple set of rules to follow. See the How To section for full details.

When complete you should have a .TAP file, ready to load into your emulator, or for copying to a DivIDE device or to convert to an audio file to be used on a real machine.

The emulator will only work on +2A/+2B/+3 machines though.

Once loaded the game is ready to play.

This isn't a clone, this isn't a version of, this IS the real arcade game running on your Spectrum. Wow!

Because of the screen aspect ratio, the scoring panel has been moved to the right hand side to allow the game area to be larger, and this works well.



The sounds are slightly different because the Speccy is limited, but overall this is a marvellous achievement. A full arcade game on a Spectrum.

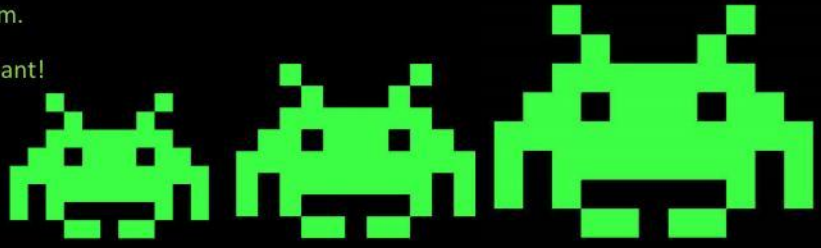
As well as being able to generate a TAP file, the program also comes with a tool to generate a DivIDE compatible TAP file. If you want to use the game via a DivIDE, run GENSI_DIVIED.EXE instead of the one mentioned in the How To section.

Because this is emulating the arcade game, you also have access to the dip switches (via software) to change things like extra lives and the tilt mechanism.

Brilliant!

HOW TO...

1. Unzip the ROM files into a folder on your computer. There should be 4 files.
2. Place the emulator files into the same folder.
3. Run the GENSI.EXE file. When prompted press RETURN.
4. A new file will now be created called SI.TAP.



GOBBLEMAN

Artic Computing 1982

When the Spectrum was released, way back in 1982, like so many other systems at the time, the first games to be available were arcade clones. This helped to get the gamers on board, take them away from the dark arcades and into the shops spending money.



Games companies were in their infancy with many being one-man outfits churning out games from their bedrooms. A lot of the games were in BASIC, but the public still bought them because there was nothing better.

One of the very first games I bought was Gobbleman from Artic Computing. As you can probably tell from the title and the inlay, this is a Pacman clone. The inlay for all early Artic games were, in my opinion, very good, and this one still looks great.

Onto the game then, and this 8k program tries to replicate the arcade, but because sprites were still not common place on the Sinclair machine, the author opted to use 8 x 8 pixel character blocks and user defined graphics.

Although they look similar to the arcade, because of the size, they don't have the same attraction. Movement too is in eight pixel jumps rather than single pixel smooth transition.

The maze is much larger, taking up the full screen, consisting of the usual blue walls and red dots with power pills at each corner. There are no gates on each side that take the player to the opposite side of the maze though.

The ghosts are less intelligent or individual than the arcade, as you would expect from an 8k game, but do the job well, offering decent gameplay.

Sound is limited to a blip when a dot is eaten and a death sound, again just blips. The death routine of the arcade, where our little yellow friend rolls back on himself is missing, replaced instead with random ascii characters.

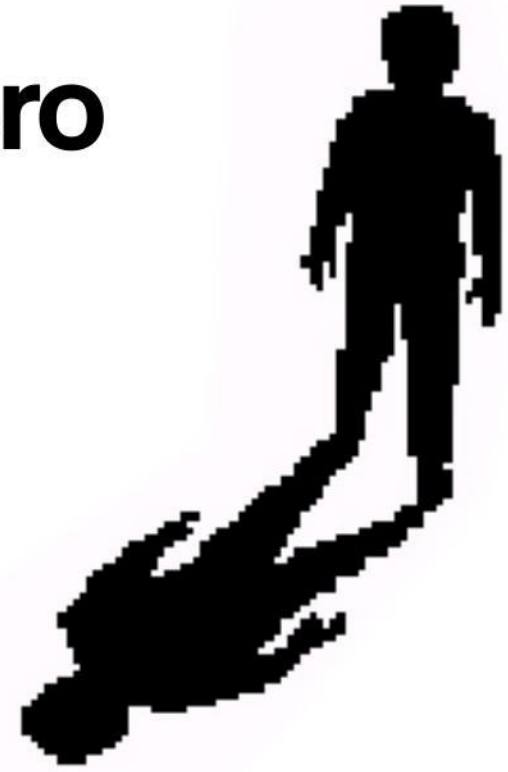
The bonus items are also missing, meaning it's just a dash around the maze.

Control is fine, and gameplay delivers a basic version of the game that can be challenging. There are problems though, and from the very first few plays back in 1982, the game crashed with an OUT OF SCREEN error. During my playing for this review though I couldn't replicate this.

So what we have here then is a run-of-mill arcade clone that has most of the elements, but due to the limitations of the machine and the knowledge of programming at the time, provides a kind of minified version. You do get a decent game out of it if you don't mind the jerky graphics.

Making Code Zero

The full un-abridged version—Part 1



If you got hold of the Crash 2017 Annual, you would have seen that I wrote a piece on this same subject, however, space was limited and I was restricted to 1500 words. This is the full, un-abridged version, that goes into more details and includes more images.

Anyone who watched the anniversary episode of the show will have seen a few of my BASIC games from the 1980's. Amongst them was a game called Countdown. This was not linked to the UK television show of the same name, but a race against time to stop a nuclear reactor from exploding.

To do this you have to locate door activation cards and use them to open the doors across five floors, all accessed by a lift.

The game used large graphics to represent the main character, 6 characters high and 2 character wide. This took a lot of UDG's and I utilised a program called Paintbox that allowed you to save out and use 4 sets of UDGs.

The game was average I suppose, and the graphics were a little odd, especially the arms of the main character. I did like the lift sections though.

In late 2016 I began tests to try and convert the game using Arcade Game Creator. There would be several challenges to doing this, things I had not done with AGD, or knew if they could be done. I decided to tackle these first rather than do a lot of work only to find the game could not be finished.

The major elements to try first were collecting and using of cards and of course the lift.

The Lift

The lift had to present a view of all floors and allow you to move between them, placing your character on the floor selected. The BASIC version worked really well, but now it was time to try it out in AGD.

I build a few graphics to represent the building and lift, along with a first attempt at a main sprite. I wanted to see how it looked before even starting the coding.

Happy (for now) with the lift screen, the next job was to get it working. Not only moving, but stopping at the floors and allowing the game to switch to the required screen.

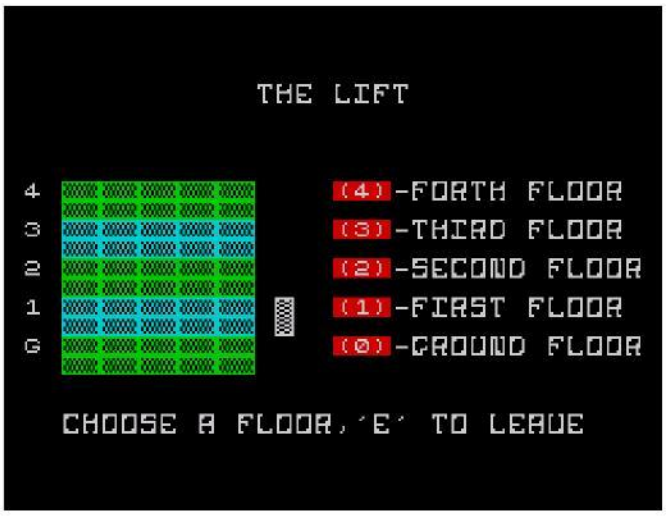
I set a variable to indicate the floor number, and upon pressing the up key, the lift would move a set number of

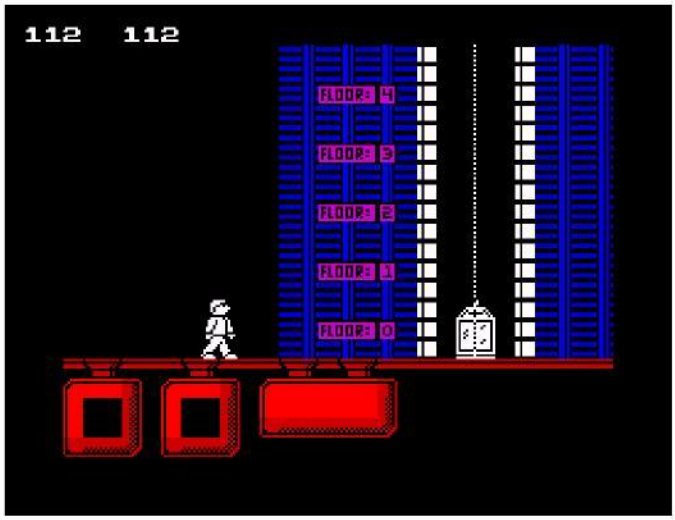
Countdown. The BASIC version from 1987. Large main character and lots of pipes.



The Lift

BASIC version on left, very early AGD version on right. Note the different main sprite.





pixels to the next floor. I added a check so it didn't fly out of the roof, and did the reverse for downward movement, and was fairly happy. Now the lift moved in both directions. Next, getting out.

Because I had the variables for the floor number, I just checked the position value and then set the screen to the desired number. This threw AGD to that screen and displayed it. Easier than anticipated apart from having to have the main sprite on screen to get the controls to work.

To hide him I cheated. I create a black paint and black ink area, blocked it in with black wall blocks and dumped him there. If you press left or right in the lift screen, you will hear him walking!

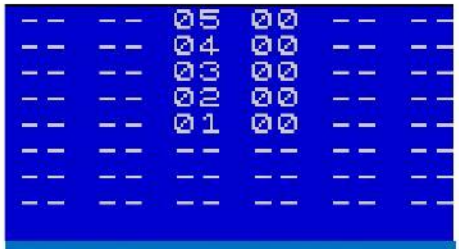
With the lift working, the next task was to get the main character looking better. I wanted a large sprite, so used this option in AGD. It uses up more memory, but looks better. I also wanted a lot of animation frames, and have him breath when stood still. A bit like the guy from Metal Slug.

A few hours of trying different things, the final image was complete, and looking quite nice. I now had to make the animation for walking left and right.

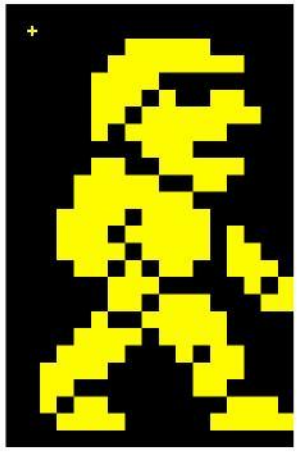
Transporters

Another feature of the BASIC game was teleporters. Because I didn't know how to make the character climb ladders, or even detect if there was a ladder there, I opted to have teleporters. This transported the character up or down the adjacent floor. Yes, it was a cheat, but it worked. I now had to get this working in AGD.

Despite spending many hours on this element, I could not get the right affect. I wanted a kind of Star Trek effect, but failed to even get the character to move platforms. I think the code was so screwed up after trying different things with different variables, I went back the saved version before I started, and decided transporters were a very bad idea.



The map screen in AGD shows how the lift works. Room 00 is the lift, floors 1 to 5 as the first rooms for each floor. The rest of the rooms for each floor will be build to the left of these 'starter' rooms.



Making Code Zero

Ladders. Everyone uses ladders, so let's go with that. But how could I make them different. I then stumbled on the idea of having a side-on view of the climb. Most games had the character facing away from the player, this sometimes made the sprite get mixed in the ladder graphics. I could avoid that by showing him climb side-on.

I draw the image of the climb, then realised I didn't actually have anything to climb!

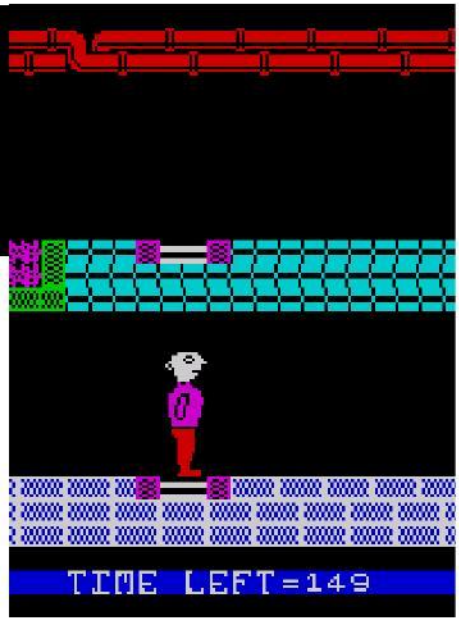
I quickly threw together some basic floor and ladder blocks to test it out, and it looked great. The only problem with this method, was that it used up more memory because you hade to have a different sprite for each direction he was facing. To get around this I cheated. I decided to design all the rooms so that he could only climb while facing left.

This would later come back to haunt me, but at this early stage, I pushed on and made the rest of the animation frames for walking, breathing and climbing.

With all that animation, I moved onto what I always enjoy, making worlds. Or at least making graphic blocks to represent areas.

I wanted the ground floor to be a basement. Dark and murky, with pipes, boilers, broken lights and the usual things you find in dingy places like that. To get this effect, I used a mainly dark blue palette, unlike the multicoloured pipes from the BASIC version.

I got a bit carried away at this point, and instead or trying to get the last game-deciding element in place (cards and doors) I got stuck into the room graphics. It wasn't long though before I stopped and began to think of ways to get the cards and doors to work.



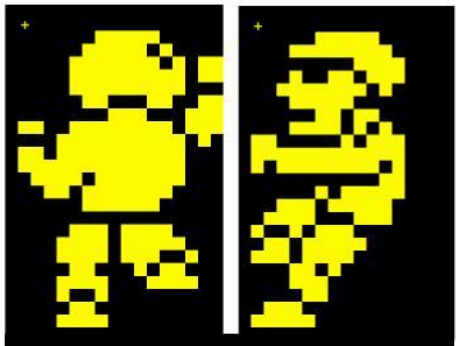
BASIC version - ready to transport. Upon pressing the transport key, a line would draw black space from the bottom of the character upward - nearly like Star Trek!

By this point I had already created the panel at the bottom of the screen after a few attempts. Although not yet used in code, I was fairly sure it would do its job. Even very early versions of the game had it.

Cards & Doors

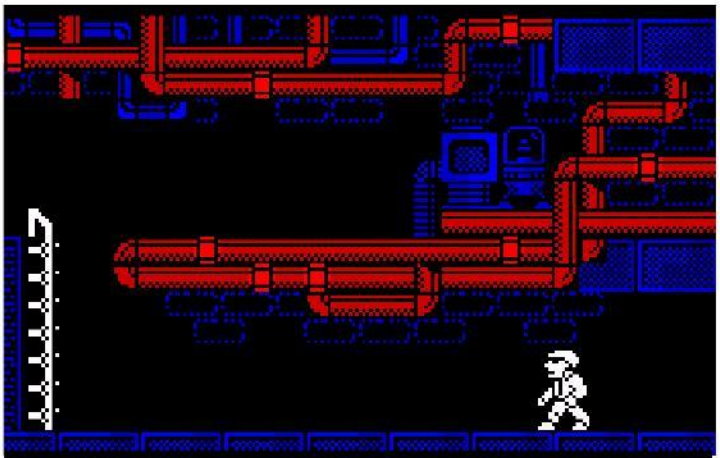
The main element of the game was the collection and use of cards. Only one card could be carried at once and each card would only open one door.

This would mean using variables to track the state of doors, cards carried and cards used. That could be a lot of variables, and AGD is limited. I also need other variables for things like animation counts, so I had to come up with a better method.



I did create a climbing animation for the usual view (character facing away), then had a brain wave and made him face left to climb.

In the end, after much head scratching, I decided to re-use variables in quite a novel way, at least I thought so at the time.



Dark and dingy! Aren't all basements like this?

First though I had to build doors and

this presented me with yet another issues. Doors can be made in different ways in AGD. I could use normal blocks, fodder blocks, objects or sprites. Each would have their own unique challenges and limitations.

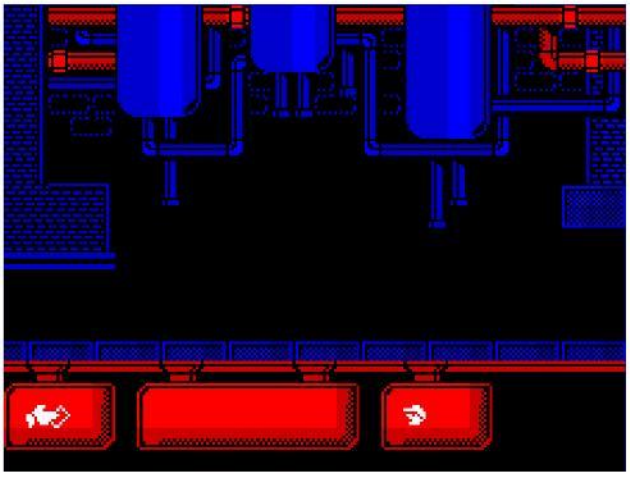
I chose to use sprites because collision detection and removal was easy, plus I could use just one sprite for every door with a cunning bit of coding.

I drew the door sprite and dropped it into a room. A few lines of code and the player would bounce back when he ran into it. Excellent, now to actually remove it.

Before I could do that though, I had the matter of the key cards to produce. These had to be drawn and code written to collect them. The player could only carry one at a time, so I used a variable to indicate if a card was being carried. I then decided to scrap that idea and instead use the idea from the original. Clever location mapping meant the player could only ever come into contact with one card at a time. This eliminated a lot of code.

The Cards

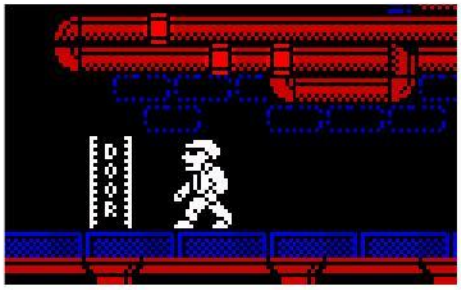
To open a door, you had to be carrying a card, and each card would only open a certain door. For this I used a single variable that held the number of the card being carried. Using this, if the player bumped into a door, a check would be done to see if he had the right card, if he did, the card and the door would vanish and the card collection variable was reset. This worked very well until I re-entered the screen to discover the door was back again. This is how AGD treats sprites, and so I had to come up with something to stop it.



Using a cunning yet simple single variable, I could easily track how far the player had got in the game, which card they were carrying and which doors to display or remove.

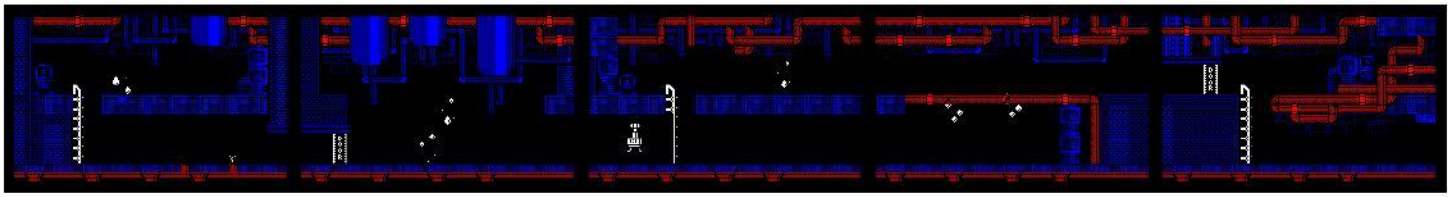
As you can see from the table below, if the variable is holding 3, you are carrying the second key, and have already opened the first door. This means entering the room with the first door would trigger the code to remove it. If you collided with the second door, the variable would be set to 4.

This variable was used for several things in the code, not only to detect which key you had and which door had been opened, but also to display or remove the door and to display the icon in the control panel indicating you were carrying a key.



Var Val	Key Held	Door 1	Door 2	Door 3	Door 4
0	0	Visible	Visible	Visible	Visible
1	1	Visible	Visible	Visible	Visible
2	0	Hidden	Visible	Visible	Visible
3	2	Hidden	Visible	Visible	Visible
4	0	Hidden	Hidden	Visible	Visible

More next issue



Nixy

The Glade Sprite



Review by André Leão





After a 2017 filled with many great games by Andy Johns (like Ooze), he returns with Nixy, a game which kept us in suspense and expecting the best for some months. And at least we can say that the long wait was fully compensated, as it exceeds even our best expectations. Everything in Nixy denotes an exemplary dedication, just starting with a beautiful loading screen, by John Blythe.

In Nixy we assume the role of a nice elf, whose task is to be the guardian of the Gaia Stone. This sacred stone is the heart of the enchanted world that gives life to plants and flowers. But something strange happened, the Stone was corrupted, and the other peaceful flowers and forest-like creatures became dangerous and attacked those approaching. The Ancients spoke of a lagoon (Pond of the Moon) that purified the stone. Maybe the solution to the problem is here? We have to travel through the forest, avoiding the plants and all kinds of deadly creatures to find the sacred pond, purify the stone, and then return it to the Guardian Tree.

Our very first task is to find the Gaia Stone. Once we find it, we will have to search for the Moon Pool, the place to deposit the Stone, as well as the ten blue blooms hidden in the enchanted world which we have to pick up. Lastly, we will have to return the Stone, already purified, to its guardian.

Does it seem a simple task? Then try playing it. The problem is not just the many obstacles we have to overcome, we also have to find all the items and places necessary to accomplish the tasks, and these are not always the most obvious. This also requires some lateral thinking as well as a great deal of dexterity in the fingers.

We start with only three lives, but in the forest, hidden, there are four more lives in magic mushrooms, just waiting



to be found. We absolutely need to find those mushrooms if we want to reach the end of Nixy, even though Andy Johns complimented the game with a generous collision system, lowering the difficulty level.

But what sets this game apart from many others of the genre? Firstly, the amazing sprites and scenery. Wandering through this enchanted forest is something marvellous, with such a profusion of colours and good taste. Then the movement of the sprites is majestic. Try to jump with Nixy and see how graceful she is and how she lands on the ground. Top score for this, improving the gameplay, something we´re already used to, in Andy Johns' games.

Finally, the magnificent music that accompanies the unfolding of this adventure, and that once again contributes for such a great pleasure playing Nixy.

If you can get to the end of this adventure you will have a pleasant revelation. We will not reveal it, so you will be even more motivated to complete the game. But at the end of the day, you will achieve some experience and begin to overcome the obstacles easier and faster. The difficulty level is adjusted so that the task is not too easy, but also not frustrating.



The physical version is on sale via Monument Microgames. Any collector should not miss this release, and, as usual, it comes packed with loads of extras. But if you want to try Nixy before buying it, you just have to go to Andy Johns' webpage and download this and the other games of this talented coder.

ULA Plus



There are two versions of this game, one for normal Spectrums and one for ULA plus. ULA plus can be set in some emulators and this produces more colours.

The designers of games have the option to then set an extended palette resulting in much more colourful games.

Nixy is one such game and using an emulator with ULA plus enabled will give a lot more colours. See the example below and compare to other screens in this review.

PLANET OF DEATH

Artic Computing 1982

"Adventure A: Planet of Death" is a text-based adventure game published by Artic Computing for the 16k ZX Spectrum. The game is the first in a series of eight such games, prefixed by Adventure A, Adventure B, and so on. The game was re-released by Sinclair Research, as part of its software catalogue - with code G14/S - and that version is memorable for having impressive artwork on the cassette inlay.

In Planet of Death, you assume the role of a lone space pilot stranded on an alien planet. The aim of the game is for you to find and repair your spaceship, so that you may escape from the planet unharmed. To do this, you must solve a series of puzzles, to overcome various obstacles on your way.

The game is an early example of the now familiar text adventure, in which the game world is presented to you as a series of locations, each with a short description, and with some containing potentially useful objects for you to collect and use. You interact with the game by typing simplified English commands at a prompt. The game engine recognises two-work commands consisting of a verb and noun—commands such as 'TAKE KEY' or 'GO NORTH'. The format of commands is very strict: unlike later text adventures that would let you type more complex commands, such as 'OPEN DOOR WITH KEY', in Planet of Death you type 'USE KEY' (and the target of use is inferred by the game engine). However, the game engine, is at, least helpful. If you perhaps encounter a locked door and enter 'OPEN DOOR', the game engine will respond 'HOW?', at which point you can enter something like 'WITH KEY' to try to unlock it.

```
I AM IN A DENSE FOREST
THERE IS A ROPE HANGING FROM ONE
TREE
OBVIOUS EXITS ARE SOUTH AND WEST
TELL ME WHAT TO DO
GET ROPE
I CANT DO THAT YET
TELL ME WHAT TO DO
USE FLINT
IT HAS FALLEN TO THE FLOOR
TELL ME WHAT TO DO
GET ROPE
OK..
TELL ME WHAT TO DO
INVENTORY
I HAVE WITH ME THE FOLLOWING:
A ROPE
A PIECE OF SHARP FLINT
TELL ME WHAT TO DO
```

Planet of Death was actually originally a ZX80/ ZX81 game and the ZX Spectrum version is little changed from the original. In particular, text is written in capitals, plus there is no use of colour nor sound. The main change seems to be the addition of a Save option - the original ZX80/ ZX81 game had to be completed in one sitting, without wobbling the RAM pack! The location descriptions also look to have been changed a little, though it is not clear the new descriptions are any better/ longer than the originals.

Being developed for the limited memory of the ZX80/ ZX81 (usually no more than 16 kilobytes could be assumed), Planet of Death is a relatively short adventure and has only brief location descriptions. However, in spite of this, I found Planet of Death to be both atmospheric and enjoyable to play, with puzzles that have a difficulty level well-suited to someone less familiar with the genre (like me) and a reasonably 'intelligent' game engine. Despite their brevity, the location descriptions are well-written, providing suspense and interest, plus the limited vocabulary is easy to use once you become familiar with the main command.

I actually played this game, first, on the ZX81 and I'm conscious that my enthusiasm for the game may stem from that. The game stands up well against other ZX81 titles, though someone starting out with the Spectrum version may be disappointed at the level of complexity in the game engine and the overall sparseness of the presentation. There are also some quirks, which I think can be attributed to the early

nature of the game. To my mind, not all of the descriptions are particularly reminiscent of an alien planet, and could equally well relate to an Earth-based adventure: Places such as an old shed and a damp cave (with cave art) don't seem particularly sci-fi. Also, there are a couple of occasions when the plot seems to jump forward and it's not quite clear how you got where you got to (though I'll avoid noting specific examples, for fear of giving away part of the solution). For me, these quirks do not detract from the game, but add to the charm (remember I may be implicitly thinking about game-playing on the ZX81).

An experienced text-adventure player may well find this game too easy. However, for the more casual retro-gamer, this is a good taster of text adventures. With a bit of patience, you will probably get to the solution with just a couple of hours of playing - though there are a couple of points where I got stuck and only progressed through luck/ dogged determination - that is, hitting everything or trying to use every combination of objects I could think of. A better strategy might be to type 'HELP', which produces useful hints for the difficult bits!

If you have a ZX80/ ZX81 system, I'd actually encourage you to play that version instead, so you can marvel at what was possible on Sir Clive's earliest microcomputers. Otherwise, you can download the Spectrum version from the World of Spectrum website.

Review by George Beckett

WHICH COVER IS BEST?

Planet Of Death originally had an inlay designed by Artic Computing, but the later release by Sinclair had a totally different design. Which do you like?





ON THE RUN

Design Design 1985

A chemical experiment has gone wrong, affecting a huge area that has seen animal and plants mutate. The danger is still not over though and there are still six canisters of deadly material that needs to be removed.

Which idiot would take on such a task, meet Rick Swift, a trouble-shooter hired by the Defence Department for dangerous missions.

Rick has been provided with a special suit, complete with jetpac and laser gun with which to survive long enough to collect all six containers.

There is a time limit too, he has to do all this in just one hour.

The game begins and Rick, a large non-animated sprite can slide (I would say walk, but there is no animation) or fly around the massive maze.

The maze is populated with a variety of plants and mushrooms. In fact mushrooms seem to be important to the creators as the inlay mentions them quite a lot. I wonder if there was an recreational medicine being used at the time?

Using similar controls to Jetpac, a game that surely must have influenced this one, along with Sabre Wulf, another Ultimate Play The Game, Rick has to navigate the deadly maze in search of the six canisters.

The mutated plants and animal come in all shapes and sizes, and if Rick comes into contact with them, he looses some health. Luckily though, there



are health pickups thoughtfully left laying around.



The graphics are large and very colourful, and everything moves smoothly enough.

Sound is used well, and the jetpac effect is really great. Other sounds are used for firing, collision and collection.



There is nothing new about the game, it is like every other maze game including the one it plagiarises, the aforementioned Sabre Wulf. In fact the whole game and inlay seem to have been created just to make fun of Ultimate, and the list of things the game does not contain, confirms this.

A decent game if you like mazes, not too difficult, but the size of the maze and the narrow passages make it a challenge.

KOKOTONI WILF

Elite Systems 1984

A great wizard by the name of Ullrich discovered that long ago there was once a powerful amulet, the Dragon Amulet. So powerful was this piece of jewellery, that it was broken in small fragments and scattered through time.

Ullrich was a bit old by the time he discovered this, so he sent his young assistant Wilf on the quest to recover all the pieces. This meant sending Wilf into different time zones, and so begins the game.

The first zone is prehistoric, and here we see Wilf having to dodge dinosaurs and all manner of creatures in his crazy quest.

This is a collect-em-up game, but it isn't the usual platform variety. Wilf can fly by using a pair of wings thoughtfully strapped to his back. This means he can fly or walk to gain access to different parts of the screen.

The obstacles in this first zone include the extremely long tongues of some dinosaurs, clouds, birds and bats.

The graphics are very nice, especially for an early game, and the large dinosaurs look excellent. The smaller mobile enemies look less impressive though and the scenery consist of solid blocks of colour.

Each time zone changes the landscape and enemy sprites, and we find Wilf having to travel through 1066, 1467, 1784, 1984 (that features Miner Willy sprites) and finally 2001 (that features a Miner Willy graphic!)



Wilf himself, moves quite slowly, so it's not a fast paced game, and the sound is a bit on the poor side. Whichever level you get to the gameplay is just the same; move, avoid and collect.

It's not a terrible game, and will pass a few minutes if you like slow moving collect-em-ups.







MIND YOUR LANGUAGE

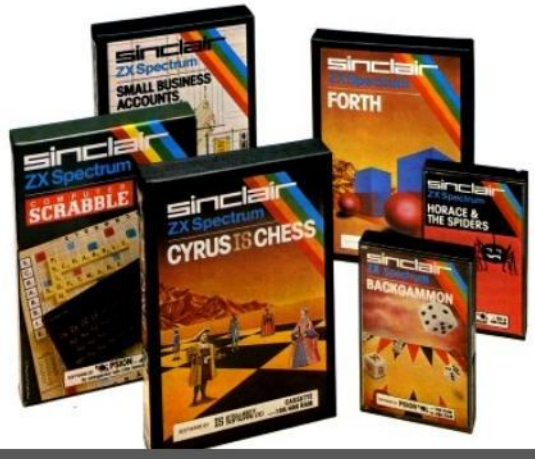
George Beckett continues his voyage through Spectrum programming languages

FURTHER FORTH

In the previous article, I introduced the FORTH programming language and suggested options for learning FORTH on the ZX Spectrum. We looked at a new version of Zeller's Congruence, written using the Artic FORTH compiler, and saw some of the advantage and quirks of the language. In the 21st century, FORTH has limitations (for example, around data handling and file I/O) which make its usability questionable. However, for a 1980's microcomputer, it provided a rare mix of performance and compactness in a (moderately) high-level language and, as such, was a viable alternative to machine code.

One particular weakness for FORTH on the ZX Spectrum is the programming environment. It can be time-consuming and cumbersome to write a FORTH program on the Spectrum, because of the screen-based view of storage and the limited editing commands. This is okay for short programs that fit on one or a couple of screens. However, for bigger projects, you may end up spending lots of time reorganising your code across screens. Debugging a program is also potentially challenging: adding in temporary code to check stack values, variable assignments, etc. is non-trivial using a line editor. On several occasions, I have inadvertently introduced bugs by overrunning the length of a screen line (at which point Artic FORTH silently drops all subsequent characters).

Despite these issues, I wanted to test the potential of FORTH for a larger programme and so I set out to write a new version of Rabbit Run. For this, I switched from Artic FORTH to Abersoft FORTH. While both compilers are based on the same dialect of fig FORTH, there are several advantages in Absersoft's compiler that make it more suitable for a Rabbit Run port. Most notably, it can hold 12 screens in memory at a time, whereas Artic FORTH can only hold one. With Artic FORTH, each time you need to move from one screen to another, you need



to flush your existing work to tape and then load the next screen from tape. In an emulator, this is a fussy exercise in tape navigation. If running on a real Spectrum, a reliable tape counter might be the only thing between you and insanity! In contrast, Abersoft's capacity for twelve screens is sufficient for most reasonable projects, including my attempt at Rabbit Run.

To cut a long story short, I have managed to produce a FORTH version of Rabbit Run that is comparable in performance to the machine-code version introduced earlier in the series. The listing of the program, which occupies 10 screens, is given in Figure 13. I have tried to adhere to good-programming practice, as described by the Forth Interest Group [http://www.forth.org/forth_coding.html], and to provide reasonable comments within the constraints of screen capacity. In the process of writing the game, I learned quite a lot about FORTH and picked up tips and tricks about the compiler, which will be useful in future projects.

Most important, it is critical to invest enough time to properly define the structure of your program before doing any coding. This is true for any programming language. However, it is particularly important with FORTH. I (probably like many others) am often guilty of moving to the keyboard too soon, sketching out the eventual program structure in code, and then filling in the functionality as I go—often having to correct (or, refactor) the high-level structure as I refine the design. With FORTH, this approach is almost certainly destined to fail—or, at least, involve a lot of retyping of code.

As for most languages, FORTH relies on a top-down approach to program-design. However, in contrast, the implementation of a FORTH program is typically a bottom-up process, starting with low-level functions and then combining these to make higher-level elements later on, eventually arriving at a word definition that encapsulates the whole program.

The structure I have chosen for the Rabbit Run is shown in Figure 11. Once implemented, the whole program will be encapsulated in the Forth word RABBITRUN. At the top level, there are two phases to the program: there is an initialisation phase called INIT, in which the game state is reset, user-defined graphics are defined, and the screen is set up for a new game; and a game loop, wrapped in a BEGIN-UNTIL loop, which runs through the tasks associated with each cycle of gameplay and which repeats until the player runs out of lives.

The initialisation and game-loop phases are each divided into a sequence of activities required to complete the phase. For example: INIT defines the UDGs in INITUDG, resets the score, number of lives, etc. in NEWGAME, clears the screen and resets colours in INITSCREEN, and prints the score and number of lives (PRINTSCORE); INITUDG is further broken down into four uses of UDGSET—a word that will copy a block of eight bytes into the right UDG space; and so on.

```
: UDGSET ( B7 ... B0 ADDR --- ) ( POPULATE USER-DEFINED GRAPHIC )
  8 0 DO ( FILL 8 ROWS OF PIXELS )
     DUP 1+ ROT ROT C! LOOP ; ( COPY + INC ADDR, WRITE BYTE )
```

Fig.1

The definition of UDGSET (fig.1) is a good example of the typical FORTH programming style, as recommended by the Forth Interest Group.

Each word definition should begin with a comment that describes the state of the (data) stack before and after the word is executed. The --- represents the execution of the word. It is preceded by a description of the stack before the word is executed: in this case, containing eight stack entries for the pixel pattern of the UDG and the address at which the UDG is stored. The number on top of stack is nearest to the ---. After the ---, a description of the stack when the word has been executed is provided: in this case, it is blank—nothing new is added to the stack, and the UDG address and pixel pattern have been removed. Following the data-stack comment, a second comment can provide a brief explanation of what the word does, and then comes the body of the definition.

The FORTH Interest Group recommends different levels of the structure be indented by multiples of three characters, to make the program more readable. In this case, the body of the definition is indented by three characters and then the kernel of the DO loop is indented by a further three characters. In-line comments are best placed at the end of each line.



FURTHER FORTH

Because each screen contains only 16 lines, it is usual to include sequences of words on each line, as above. This ensures memory is used more efficiently - a screen line occupies 64 bytes of memory irrespective of how much FORTH code it actually contains.

The Forth Interest Group also recommend that individual word definitions do not extend beyond a single screen, breaking down any longer definitions into smaller units. I have tried to follow this advice, with the exception of MOVEFOX and RABBITRUN, for which I decided there was no obvious way to decompose them further.

My design for Rabbit Run is based on having the same stack layout throughout most of the game loop, containing three important values: the iteration number is the third stack entry and the coordinates of the rabbit are the second and first stack entries. A stack depth of 3 is very manageable - it is quick to retrieve any of those entries to the top of stack using words such as ROT and OVER—whereas a deeper stack would require more work to manage. I have elected to keep other, less time-critical elements of state - such as the score and location of the fox, in variables. Variables are slower to access than stack entries, but by keeping the stack small, few stack manipulations are required, so the game loop is fast.



At the beginning of the program, I have defined four constants that control how the game behaves. This makes it easy to change the speed and difficulty of the game during testing. In FORTH, constants are inserted at compile time, so it is no slower to use a constant such as FOXAPPEARS than to use a literal value such as 100. After the constants, the game variables are defined. In Forth variables are global: they can be accessed from anywhere, once they have been defined.

After the variables, the body of the program follows and, as noted above, this is written in a bottom-up order. Higher-level words cannot be defined until any lower-level words they depend on have been. In the program structure, in Figure 11, this means that words at the ends of branches must be defined before higher-level words can be tackled. Hence in Rabbit Run, the first words to be defined are the low-level functions such as a random-number generate RND and a sign function SGN (neither of which are part of the standard FORTH vocabulary).

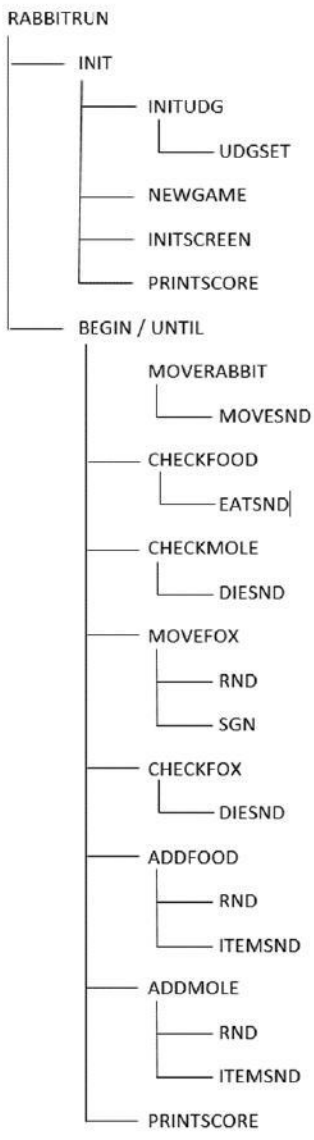
Further down the listing, higher-level words are defined until we reach the top-level definition of RABBITRUN.

As noted above, I am relatively new to FORTH, so it is likely a more experienced programmer could write a better implementation than I have. However, despite my inexperience, the game works and runs smoothly. It is comparable in performance to the assembly language version, but took me significantly less time to develop—even though I have been writing Z80 assembly language for more than 30 years.

The only significant drawback of the FORTH version compared to the assembly-language version relates to reading the keyboard. As with BASIC, FORTH can only easily read one key-press at a time. This means that the rabbit can move horizontally or vertically, but not diagonally. You may have noticed that it was possible to move diagonally in the machine-code version, which made the game easier to play.

If you study the code closely, you may notice I have done something apparently odd in the FOXMOVE routine. The next fox move is computed irrespective of whether or not the fox is in play. Then, at the end of the function, the working is discarded if the fox is not yet on-screen. This may seem a waste of time, but I chose to do this to ensure there was no noticeable change to the timing of the game, when the fox appears. FORTH is fast enough that I can include these unnecessary calculations and still have a fast-paced game.

One issue with the versions of FORTH considered so far is that they only support tape storage. Given the organisation of programs into screens, it can be quite frustrating to write longer FORTH applications. There do appear to be FORTH compilers for the Spectrum that support the ZX Microdrive. However, the ones I found are listed as 'Missing



in Action' on the World of Spectrum website and I have not been able to find any information about them.

Because of this, I looked to the CP/M Plus operating system on the ZX Spectrum +3 (you may recall Mallard BASIC for CP/M Plus from an earlier article). CP/M software is still reasonably accessible, via websites such as www.cpm.z80.de. On that site, I found a port of fig FORTH, which I copied onto a +3 disk image using the CPCDiskXP utility [www.cpcmania.com].

```
 1 : TASK ;
 2 : ZELLER ( DAY MONTH YEAR )
 3 >R
 4 DUP 3 <
 5 IF 12 + R> 1- >R THEN
 6 1 + 13 * 5 / +
 7 R> 100 /MOD
 8 21 * 4 /
 9 SWAP
10 5 * 4 /
11 + +
12 7 MOD ;
13 : SQRT-CLOSER 2DUP / OVER - 2 / ;
14
15
ok
7 LOAD R MSG # 4  I MSG # 4  ok
EDIOR EDIOR? MSG # 0
EDITOR ok
14 P : SQRT 1 BEGIN SQRT-CLOSER DUP WHILE + REPEAT
1 LOAD TASK MSG # 4  ok
9 SQRT . 3 ok
25 SQRT . 5 ok
```

The fig FORTH compiler works well on the +3 provided you set the display to 80-column mode (for example, by running the SET24X80 command). In fact, it turns out that the CP/M version of the compiler is very similar to the Abersoft version, and is ported from the same source code. The two compilers have the same editor vocabulary and many standard word definitions in common.

There are several downsides to the CP/M compiler. First, it is written for a generic CP/M platform, so does not support the Spectrum's hi-res graphics nor, as far as I can tell, colour and sound; so it is only suitable for text-based applications and games. The other and more significant downside is that I could not find any proper documentation, other than a short README file included in the distribution and a scanned installation manual from the Forth Interest Group [www.forth.org]. Therefore I have mostly had to rely on trial and error.

The distribution includes several useful FORTH source codes, most notably a file called SCREENS.FRT, which has a text editor, a machine-code assembler, and some utilities for serial communications.

To use one of these source files, you start FORTH with a command such as 'Z80FORTH SCREENS.FRT'. As fig FORTH does not understand a high-level file system, the CP/M port maps FORTH's view of block storage (that is, screens) onto a suitably structured file. Effectively, the FRT file is presented to fig FORTH as if it were a block storage device.

You can explore the contents of the block storage using INDEX, which will display the first line of each screen in the range specified—e.g. 1 10 INDEX will display details of the first 10 screens. The editor is stored in screens 7 to 12 of SCREENS.FRT, and can be loaded using 7 LOAD (followed by EDITOR to enable the editor vocabulary). A list of the available editor commands can be found in a file called EDITOR.DOC included in the distribution: as noted above, they are the same as the editor commands in Abersoft FORTH.

If you use the editor as described above, then your changes will be written into the SCREENS.FRT file, so I recommend you make a copy of the SCREENS.FRT file first (for example, using the CP/M PIP command). If using standard single-density 3-inch disks, you will probably need to copy the file to a new disk, as the SCREENS.FRT takes up a lot of space. You will also need to copy the main program Z80FORTH.COM, if you want to run FORTH from the new disk.

As provided, there are not many free screens in SCREENS.FRT. There is a

block of five empty screens, numbered 41 to 45. However, you can create more space, either by clearing screens that you are not interested in—for example the modem code, which occupies screens 18—39, or you can use EXTEND to add more screens to the end of the source file.

Some more research is needed to get the most from the CP/M port of fig FORTH. The web does not seem to have much useful information, so one may need to post questions on specialist forums or even look into the source code. However, with the knowledge gained from this and the previous article, it should be possible to write some interesting and useful programs.

```
SCR # 1

0 ( RABBIT RUN )
1  30 CONSTANT MOLEFREQ ( TIMESTEPS BETWEEN MOLEHILLS )
2  47 CONSTANT FOODFREQ ( TIMESTEPs BETWEEN FOOD )
3   4 CONSTANT FOXFREQ ( TIMESTEPS BETWEN FOX MOVES )
4 100 CONSTANT FOXAPPEARS ( TIME WHEN FOX APPEARS )
5 0 VARIABLE SCORE ( PLAYER SCORE )
6 3 VARIABLE LIVES ( LIVES LEFT )
7 1 VARIABLE FOXROW ( LOCATION OF FOX )
8 1 VARIABLE FOXCOL
9 0 VARIABLE OFOXROW ( PREV LOCATION OF FOX )
10 0 VARIABLE OFOXCOL
11 0 VARIABLE FOXFLAG ( INDICATES IF FOX IS ACTIVE )
12 -->
```




```
SCR # 2

0 ( USER-DEFINED FUNCTIONS )
1 : 1- ( N --- N-1 ) 1 - ; ( DECREMENT )
2 : RND ( N --- RND_N ) 23672 @ SWAP MOD ; ( RND NUMBER GENERATOR )
3 : UDGSET ( B0 ... B7 ADDR --- ) ( POPULATE USER-DEFINED GRAPHIC )
4   8 0 DO ( FILL 8 ROWS OF PIXELS )
4     DUP 1+ ROT ROT C! LOOP ; ( COPY + INC ADDR, WRITE BYTE )
5 : SGN DUP 0 > IF DROP 1 ELSE 0= IF 0 ELSE -1 ENDIF ENDIF ;
7 ( GAME SOUNDS )
8 : MOVESND 1 1548 BLEEP ; ( RABBIT MOVE )
9 : ITEMSND 2 759 BLEEP ; ( FOOD OR MOLEHILL APPEARS )
10 : EATSND 4 412 BLEEP ; ( RABBIT EATS FOOD )
11 : DIESND ( RABBIT DIES )
12   12 1763 BLEEP 11 1959 BLEEP 10 2202 BLEEP 9 2514 BLEEP
13   7 2926 BLEEP 6 3498 BLEEP 5 4345 BLEEP 4 5727 BLEEP
14   3 8383 BLEEP 1 15595 BLEEP ;
15 -->

SCR # 3

0 ( INITIALISATION ROUTINES )
1 : INITUDG
2   60 102 126 90 126 60 66 66 UDG UDGSET ( RABBIT )
3   0 60 126 126 126 126 60 0 UDG 8 + UDGSET ( MOLEHILL )
4   0 24 24 36 36 66 66 0 UDG 16 + UDGSET ( FOOD )
5   0 24 60 60 60 102 102 0 UDG 24 + UDGSET ; ( FOX )
6
7 : NEWGAME
8   0 10 16 ( TIMER AND RABBIT COORDS )
9   0 SCORE ! ( RESET SCORE )
10   3 LIVES ! ( RESET LIVES )
11   0 FOXFLAG ! ( FOX INACTIVE ) ;
14 -->

SCR # 4

0 ( INITIALISATION ROUTINES, CONT'D )
1 : INITSCREEN
2   1 BORDER 0 PAPER 7 INK CLS
3   0 2 AT ." SCORE" 0 16 AT ." LIVES"
4   8 PAPER 8 INK ;
5
6 : PRINTSCORE
7   0 8 AT SCORE @ . 0 22 AT LIVES @ . ;
8
9 : INIT
10   INITUDG ( SET UP GRAPHICS )
11   NEWGAME ( RESET GAME STATE )
12   INITSCREEN ( RESET SCREEN )
13   PRINTSCORE ( SCORE AND LIVES )
14   2DUP AT 7 INK 144 EMIT 8 INK ; ( PRINT RABBIT )
15 -->
```

```
SCR # 5

 0 ( MAIN GAME : 1 )
1 : MOVERABBIT ( KEY --- )
2   CASE
3     80 OF 1+ ENDOF ( RIGH
4     79 OF 1- ENDOF ( LEFT
5     65 OF SWAP 1+ SWAP E
6     81 OF SWAP 1- SWAP E
7   ENDCASE
8   1 MAX 30 MIN ( CHECK X
9   SWAP 1 MAX 20 MIN SWA
10 : CHECKFOOD
11   2DUP ATTR 4 = IF
12     7 INK 2DUP AT 144 EM
13     EATSND SCORE @ 1+
14   ENDIF ;
15 -->

SCR # 6

0 ( MAIN GAME : 2 )
1 : CHECKMOLE ( --- ) ( CHEC
2   2DUP ATTR 2 = IF
3     2 INK 1 FLASH 2DUP AT
4     DIESND
5     7 INK 2DUP AT 144 EMI
6     LIVES @ 1- LIVES !
7   ENDIF ;
8
9 : MOVEFOX ( --- ) ( IF ACTIV
10   FOXROW @ OFOXROW !
11   2 RND IF ( MOVE VERTIC
12     DUP FOXCOL @ - SGN
13   ELSE
14     OVER FOXROW @ - SG
15 -->

SCR # 7

0 ( MAIN GAME ROUTINE: 3 )
1   ENDIF
2   FOXROW @ FOXCOL @ A
3   FOXFLAG @ AND >R RO
4   R> AND IF ( ONLY MOVE
5     OFOXROW @ OFOXCOL
6     FOXROW @ FOXCOL @
7   ELSE ( RESTORE ORIGIN
8     OFOXROW @ FOXROW
9   ENDIF ;

SCR # 8
```

```
HT )
)
ENDOF ( DOWN )
NDOF ( UP )

BOUNDS )
AP ( CHECK Y BOUNDS ) ;

IT 8 INK ( REDRAW RABBIT )
SCORE !

CK IF LANDED ON MOLEHILL )

T 134 EMIT 0 FLASH

T 8 INK

E, MOVE FOX )
FOXCOL @ OFOXCOL ! ( SAVE FOX LOCN )
CAL / HORIZONTAL )
FOXCOL @ + FOXCOL ! ( VERTICAL )

GN FOXROW @ + FOXROW ! ( HORIZONTAL )

TTR 2 = NOT ( CAN'T MOVE ON MOLEHILL )

T DUP >R ROT ROT R> FOXFREQ MOD NOT
IF FOX ACTIVE AND FOXFREQ )
@ AT 7 INK 32 EMIT 8 INK ( DELETE PREV )
AT 6 INK 147 EMIT 8 INK ( PRINT NEW )
AL FOX COORD )
! OFOXCOL @ FOXCOL !
```

```
0 ( GAME ROUTINE : 4 )
1 : CHECKFOX ( CHECK IF FOX CATCHES RABBIT )
2   2DUP ATTR 6 = IF
3     1 FLASH 2DUP AT 134 EMIT 0 FLASH
4     DIESND
5     7 INK 2DUP AT 143 EMIT 8 INK
6     LIVES @ 1- LIVES !
7     1 FOXROW ! 1 FOXCOL !
8   ENDIF ;
9
10 : ADDFOOD ( ADD MORE FOOD )
11   ROT DUP FOODFREQ MOD 0 = IF
12     19 RND 2 + 29 RND 2 + AT 4 INK 146 EMIT 8 INK ITEMSND
13   ENDIF
14   ROT ROT ;
15 -->

SCR # 9

0 ( GAME ROUTINE : 5 )
1 : ADDMOLE ( ADD ANOTHER MOLE HILL )
2   ROT DUP MOLEFREQ MOD 0 = IF
3     19 RND 2 + 29 RND 2 + AT 2 INK 145 EMIT 8 INK ITEMSND
4   ENDIF
5   ROT ROT ;
6
```

```
7 : RABBITRUN ( MAIN PROGRAM )
8   INIT
9   BEGIN
10     INKEY DUP 255 = IF ( 255 MEANS NO KEY PRESS )
11       DROP
12     ELSE
13       ROT ROT 2DUP AT 32 EMIT
14       ROT MOVERABBIT 2DUP AT 144 EMIT MOVESND
15 -->


SCR # 10

0
1     ENDIF
2     CHECKFOOD CHECKMOLE
3     MOVEFOX CHECKFOX
4     ADDFOOD ADDMOLE
5     PRINTSCORE
6     ROT 1+ ( INCREASE TIMER )
7     DUP FOXAPPEARS > IF
8       1 FOXFLAG !
9     ENDIF
10    ROT ROT ( RESTORE STACK ORDER )
11    LIVES @ NOT UNTIL ;
```

More from George next issue.



Sinclairs Amstrads Commodores Acorns Atari STs Amigas

repaired and refurbished

Mutant Caterpillar Games is the place to go to for **tested, working**, refurbished 8-bit and 16-bit hardware and software.

http://www.mutant-caterpillar.co.uk

PAWS

Artic Computing 1985



This game was originally named Cats, but after a legal wrangling from Andrew Lloyd-Webber, Artic were forced to rename it as Paws.

Onto the game then, and your task is to control a mother cat that finds her kittens have all strayed off while she was out gathering food. She now has to go searching for them around the town and countryside, bringing them back one at a time.

Roaming the area are vicious dogs, that will attack the cat if she can't get out of their way. To make sure she has a good chance of winning the fights, she must keep eating food and have a good amount of stamina. Stamina decreased when walking about and when she fights the dogs.

The food is scattered about all over the place in shape of fish, milk and mice. There is plenty of it too, at least on the easy levels.

There is a time limit to complete this task, which is midnight, but the screen does not indicate what time it is or give you any idea of how long you have left. She has to get all the kittens back by midnight to create enough Catoplexic energy to fend of the pack of dogs hunting them.

This is a maze game, taking a lot from Sabre Wulf, and in fact some of the countryside scenery is very much like the Ultimate classic.

The towns have various recognisable buildings, such as the TSB bank, Menzies, Woolies, Natwest, Midland and Odeon Cinema, and some areas look like a junk yards, with different game graphics forming the walls. Its all very colourful and well-drawn, but the path she has to take is very restricted to angular movement, even in open spaces, she has to follow the 90 degree angles.

If she does get into a fight, there is a short animation that reminds me of Maziacs, and her stamina and strength should see her through if they are high enough.

The sounds are limited to collecting things and fighting, there is no sound for walking, so a lot of the game is played in silence.

Once she finds one of the kittens, the main sprite changes to show her carrying it, which is a nice touch. She then has to get back to the kitty home to deposit it before heading out again to find the next one.

There are X kittens to find, and a lot of stray dogs roaming about…

This is not a bad game, especially if you like maze games and despite the controls being a bit limited, which can cause unavoidable fights, it's worth a quick play.



Jilly's Farm Volume 1

SokoBAArn

Bob's Stuff 2017

Remember Farmer Jack? He has retired and his daughter Jilly now manages the farm. After the harvest many bales of hay left on the field and there is no one available to gather it. Maybe some of the animals can help?

If you watched wonderful Aardman's animated movies, like A Close Shave or Shaun The Sheep, you know that sheep are clever animals. One of them will try to gather bales of hay.

Jilly's Farm Volume 1... SokoBAArn is a Sokoban clone with an isometric viewpoint. You control a sheep and have to move bales of hay to specified places, marked with empty cubes.

The only way of moving bales is pushing them (even a few at once). If you make a mistake you can undo the last few moves.

There are 30 levels in SokoBAArn and after completing each of them you receive a level code. There is no time limit and you can try to solve each puzzle as many times as you want. After completing all levels - which shouldn't take long because they are not difficult - you can create your own puzzles using built in editor.

This game was created by Bob Smith, author of many excellent titles like Farmer Jack trilogy, All Present And Correct, Stranded 2.5 and W*H*B. His games are based on interesting ideas and nicely crafted.

SokoBAArn proves that Bob Smith is still one of the best authors of Spectrum games. The idea is not new but isometric viewpoint makes it original and fresh.



The graphics are monochrome but very detailed and well animated. There is also music at the beginning and simple sound effects during playing (both by Jilly Walker). The sheep can be controlled by keyboard or joystick.

I enjoyed this game and I can recommend it to everybody who likes puzzle games.

Jilly's Farm Volume 1... SokoBAArn runs on any Spectrum with at least 48k of RAM.

Piotr "PopoCop" Szymanski

P.S. There are few special guests from other isometric games. Pay attention to the pictures in main menu.







RD DIGITAL TRACER

RD Labs began advertising in several computer magazines around the middle of 1982. Their advert was uninspiring and mentioned Data logging, automatic testing and interactive graphics. The small print mentioned an interface for handling analogue input and output, but how many users would be interested in that?

Towards the end of the year, they began to advertise something quite different to the usual joystick interface or storage system, a Digital Tracer.



The system consists of an interface that looks like it was built in someone's shed and is definitely a cassette box underneath all that resin. It does have LED's to indicate when things are happening though, and a through port.

Connected to this is a large plastic articulated arm, at the end of which is a cross hair. There is also a manual and of course the cassette.

Once plugged in, the first problem immediately become clear; how to anchor the arm to stop it moving about?

RD labs supplied some double sided tape to do this, but obviously thirty years later, it's not as sticky as it was. Instead of applying more tape, I opted just to hold it and see how things went.

The manual covers setting up the device along with details about the included tools and a large program for you to type in. My version has the correct and updated version as a separate sheet.

The tape comes with versions of the program for both 16 and 48k machines, and you can load each tool separately or together if you have a 48k model.

The tools include drawing, scaling, retrace – which is a kind of copy / paste tool and graphics – which is a UDG editor.

Lets get down to it then….

Once loaded you will probably get a continual beeping and a message telling you that something is out of range. You have move the tracer arm about until a cross hair appears and the beeping stops. Then you can line up the crosshair, screen position and the thing you want to trace.

Pressing C will set continuous draw mode, and here the arm will draw whatever you trace..

You can draw straight lines or circles if you wanted to but those options are not best suited to the trace arm due to positioning problems.

You can turn on and off continuous draw by pressing the C key again.

Using this simple control, you can quickly produce a rough estimate of what is on paper.

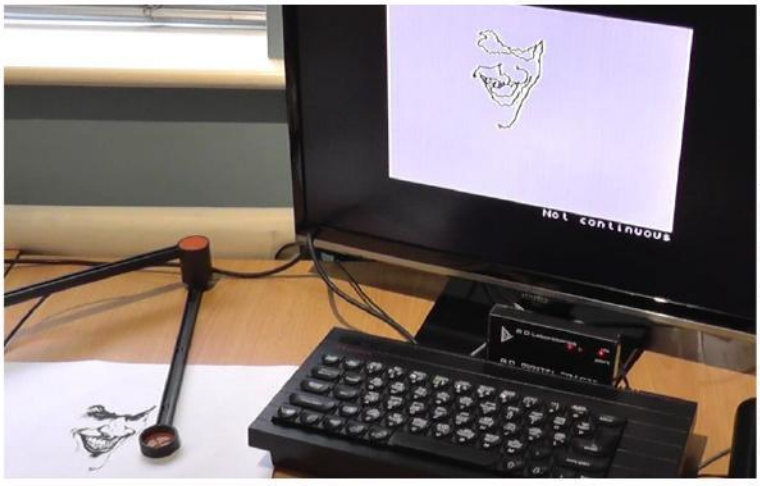
I say rough because there are many things at play that decides the final outcome.

The movement of the arm and how smooth it is, how steady your hand is, how quickly you move, just name a few, so don't be surprised if your first attempts are…. less than accurate.

Once you get used to it though, things do improve, but the re-positioning of the cross hair can sometimes be hit and miss.

I will stress again, this is not a drawing tool, it is a tracing tool, and as such there are very limited graphical commands at your disposal.

The best use of this would be to set a base line picture and then load it into an art package such as OCP Art Studio to finish it.



The retrace options just let you draw things you have already traced in different positions and different sizes, and there are some fills to use, but again,



this is best done outside of this package.

The UDG editor is rubbish and there is no reason to have it, so I'm not even going to review it.

By the end of the review my hands were aching from trying to hold the arm still and at the same time trying to keep the movement of the arm as smooth as possible. If you were going to use it for any prolonged period, you would fix it down permanently rather than trying to hold it with your hand.

The RD Digital Tracer was produced to allow conversion of paper based drawings to screen, and for that, once you get it set up, it works fine. There are issues around placement, movement and of course resolution, but this is a Spectrum after all, and you're not going to get professional results from a £50 add-on.

It was fun to play with, and I can see it's uses. An interesting bit of kit then than can produce good results if you are prepared to put the effort in.





Revival comes round fast, and this year was double the fun. Double the space, double the arcade cabs, double the consoles and computers and double the fun.

The upper hall, previously the only hall for this event, had the arcade cabs in the centre, surrounded by consoles and computers. Around the edges were the traders selling a multitude of goods for all makes and models of system.



The Upper Hall

The new addition this year was another hall, the 80's hall. Here were yet more arcade cabs, more computers and more traders.

The two day event included some excellent talks from industry veterans from the likes of Rare, Ocean, Sony, ID Software and Commodore.

If you have never been to this event, make sure you put it in your diary for next year. It's always a brilliant occasion.







The Lower 80s Hall





The Spectrum Next talk feature Simon Butler, Jim Bagley, Rich Stevenson, Mike Dailly and Lyndon Sharp. It was hosted by Dean Swain (Retro Asylum) and myself and lasted just short of 1 hour.

WARNING: MAY CONTAIN SPOILERS

GRUMPY OGRE'S Adventure Page

Welcome back to strange places where your mind wonders freely and your brain shifts into a higher gear.

My main inclination for adventure games drags me always to the earlier affairs. Things were simple, short and mostly used two words to communicate with the Spectrum.

Yes, you often had to play guess the noun, but that seemed so much more simpler than trying to guess a whole sentence and get the words in the right order.

I do try to play newer titles though, to try and find games that I may just like. One such game was Imagination from Firebird, released in 1986.

You are dragged inside a computer when you try to use a mysterious disc in it... tut tut.. A disc!

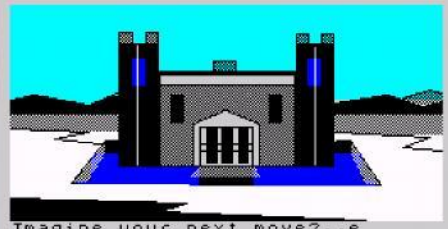
The game instructions are very bland and give no real idea of how the mechanics work, so this makes the game frustrating. The only thing you are told, is that your task is to count how many stars are in the universe.

Once you get into the game proper (by putting the disc into your computer and examining the monitor) you find there are four areas to go to. This idea is used in many games and usually consists of getting one object from one area and using it in another.

In this game the areas are all comedy names, which is a nice touch, but the game humour is a bit.. Well.. Laughable really.



I went to the space station first, and after wandering about I found a bucket, a cord and a map. I also discovered a console missing a Ramboard. Then I got stuck.



I found nothing more to do in that area, but had no idea how to get out and visit another. Eventually after about 30 minutes, I dug out the playthrough and found you have to type PINCH ARM. Why is that not mentioned in the instructions? A fairly important aspect of the game, and difficult to guess.

The game has some nice graphics, but for me there were too many problems, even in this first area.

The text had no spaces between commands and responses, making for a very hard to read game. The text itself used capitals for things like objects and important things, and lowercase for the rest. This again, made it very arduous to read.

Commands were not what I would call normal. To look around the usual command is L, but here you

have to type the word fully, LOOK.

I visited another area, a snowy wasteland with a citadel, a screaming child and a tower with a beautiful woman called Gale Wynds - the mother of all weather. Yes, that's the kind of humour to expect, and it is not really my kind!

I pushed on for a bit, but to be honest it didn't draw me in. I felt I was playing a silly game made by a friend who had passed it on for a laugh.

After more time in area two with nothing to show for it I visited area three. This tries to make a joke about Manic Miner by having the player move left across platforms (in text form) and having sprites appear and steal lives. Not sure what this adds to the game!



Next I visited area four and found myself in a World War 2 tank. Jumping outside there was a pile of things to grab, but the spade caught my eye. Way back (about an hour to be exact) I had happened across some melting snow in area two. Maybe this could be used to dig.. Off we go..

Pinch arm. Two. Blah Blah.. Dig.. Nope. What a waste of time!

Time was moving on, and the lack of progress and direction in the game left me feeling in need of some grog. I pulled the plug and stomped away thoroughly disappointed.

About the only thing I liked was a response I got when I entered SLEEP while in the sleeping quarters...

"You huddle up..ZZZZ..
you dream you are playing
Space Invaders on a ZX81.
AHHH!! a NIGHTMARE.."

Moving on...

If you are reading this in the correct order, you will not yet have reached the new section in the later pages, that covers the wonderful 16/48 tape magazine.

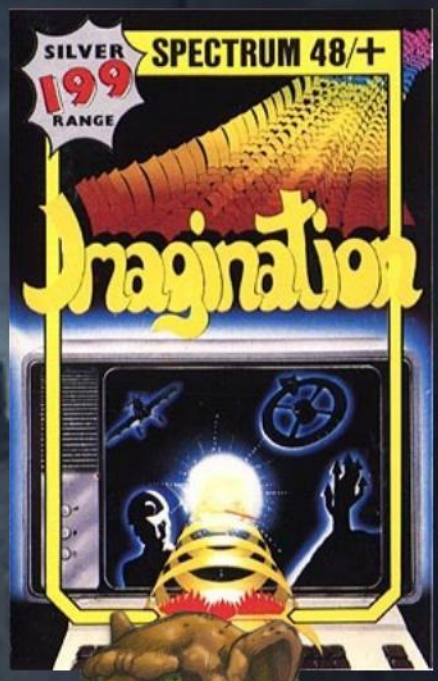
One of the highlights along with the adventure section; 'Of Dungeons And Green Men' was their serialised adventure, The Long Way Home.

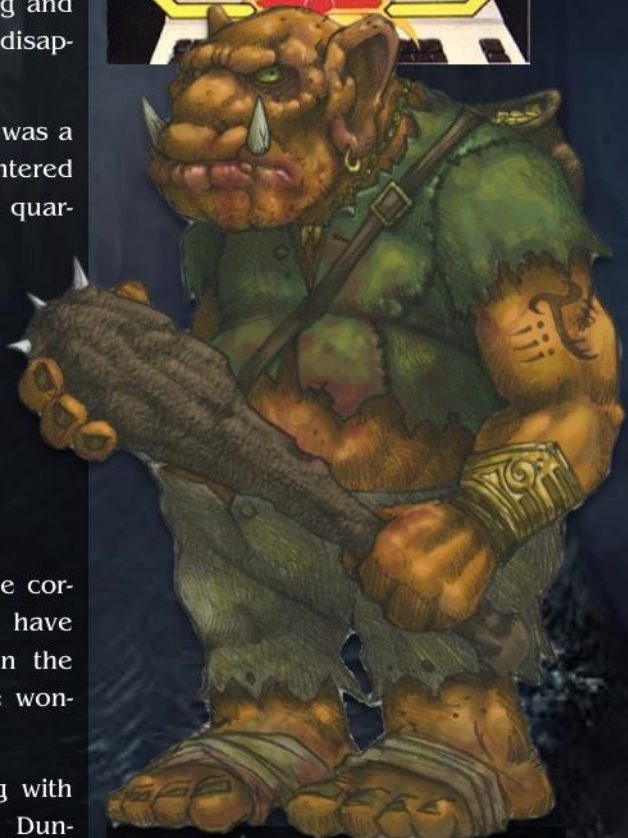
Not only was this a cunning ploy to get you to buy the next issue, but it was a nice looking game too.

In my next instalment I will revisit this and see if it was really as good as I remember it, or if it was total hog wash.

Either way, another chance to read the excellent magazine and drink more grog.

Tarra!







ZX PRINTER

I love the smell of burning paper in the morning

Many computer manufacturers provided ways for their micros to print. Usually this took the form of an industry standard serial or parallel port. Sinclair however, in their bid to keep prices low, opted for a propriety device.

The ZX Printer was launched in October 1981 and aimed at the ZX81 micro, but the design of the Spectrum's interface meant it would also work on that. It costs initially £49.99 and was heavily advertised.

This little device became the standard printer for many people. Its small form factor and ease of use meant it could be easily setup and used, and easily incorporated into any software package thanks to Sinclair embedding commands into the ROM of the Spectrum from day one.

It was simple to connect and had a through port, although it was not large enough for me to use my SmartCard and I suspect other more modern interfaces.

Loading the paper was easy, you disconnected the back, pulled the paper through, slotted it back in place and using the small pieces of plastic in the paper roll, you slotted it into the holder. It takes about 30 seconds to fit a new roll.

The sticker on the box stated you would need a separate power supply to use it on a Spectrum, but it seemed to work fine without one.

The Spectrum's ROM, as mentioned before, had built in functions to allow listing, printing direct or full screen dumps.

LPRINT was the same as print, and sent anything in quotes to the printer.

LLIST would send any BASIC program listing to the printer.

COPY would send the contents of the screen to the printer.

The device was small, measuring 14cm x 7.2cm and 5cm high. The styling suited both the ZX81 and the Spectrum, and it looks really rather neat.

It is not a thermal printer, or a dot matrix, instead it uses electricity to burn holes into special aluminium coated paper. In low light it looks cool, if a little scary as the sparks jump about.

Because it uses this method, you do

get a distinct burning smell when in use.. especially if you are printing a lot of text or screen dumps.

The noise it makes will be familiar to anyone who owned one and isn't loud enough to become distracting. It is certainly not as loud as a dot matrix variant.

Speed wise, it's not too shabby, churning out about a line every second which is ideal for listing programs you have typed in. The quoted specification from Sinclair was 50 characters a second. The quality though is not the best.

Because of the paper and method, text can sometimes be blurred or out of line and the silver look of the paper has an odd, yet quaint effect.

Using a word processor like TasWord and printing out at 64 characters per line produces a readable output, but not something you could use professionally.

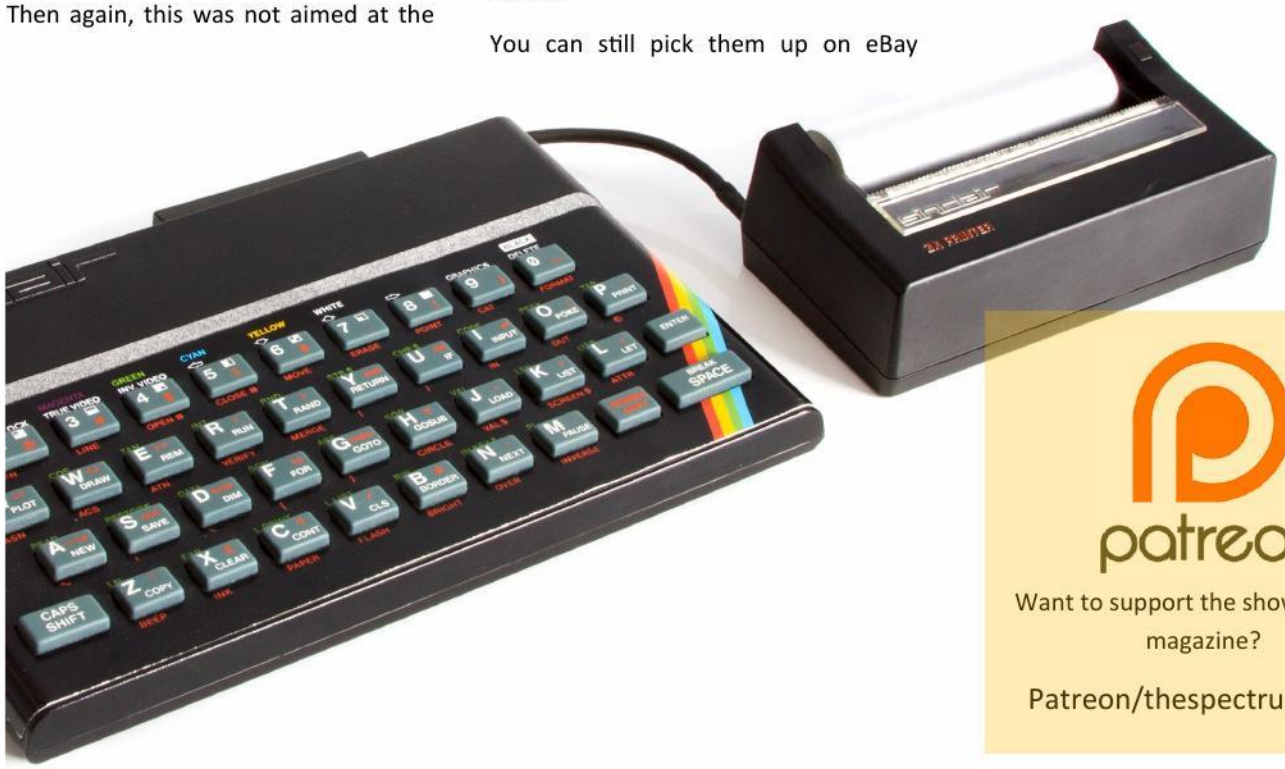
Then again, this was not aimed at the business market. It was a cheap device for home users to get output from their micro. And for this it was a nice piece of kit.



Sinclair stopped producing the unit in May 1984, much to the dismay of users. It was still a much loved and used device.

You can still pick them up on eBay although many will need a new drive belt.

I quite liked this little device. I never bought one back in the day, I chose the Alphacom 32 instead, which does produce better results, but for sentimental value, you can't beat this little printer.



patreon

Want to support the show and this magazine?

Patreon/thespectrumshow

16/48

Issue #1 November 1983



What would become a familiar sound to anyone who bought this tape magazine on a regular basis, a sound effect informs the user to stop the tape. At this point the main content listing is presents and each of the six boxes scrolls to give details about what you can expect.

There are some interesting things on the tape too, so let's get straight to it.

First we get Copter, a game. Obviously, no typing involved here, which was one of the great benefits of getting a magazine on tape. The controls though, are very odd, having different keys for left, right up, down, aim left, aim right and fire. So much to remember!

The game has two options and you can toggle between them. First is a game similar to Paratroopers, where you aim a gun at a descending helicopter. Pressing N will switch control to your Copter and you can lead the chasing enemy into a barrage balloon.

Not a bad game, but too many controls.

Next we get the editorial, and as this is the first episode, it's mainly a welcome and laying out the magazines plans for future editions. They also ask for reader contributions, offering £50 for good, original programs. They also ask for letters or articles to be sent in too.

The next program on the tape is Stroke 4. An animated diagram of a four stroke engine. You can change the speed of the animation, or pause it.

Next we get the first reviews, and these are of the DK'Tronics Light Pen and the RD Digital Tracer. The reviews consist of a few screens of text plus something created by each device. Using this on an emulator means the pictures flashes past and you don't really get to see it.

Onto side two, and one of my favourite sections, Of Dungeons And Green Men.

For those not into adventure games, the title comes from a puzzle in an early Artic Computing game involving a green man and a pair of gloves.

This section gives hints for gamers and in this edition they are for Planet Of Death – reviewed in this magazine on page 20.

There is also a hint for The Hobbit, with a promise of more to come.





At the end of the tape we get to see the contents of next months tape as a teaser to try and tempt you to save you pocket money in readiness.

After a while you are asked to press a key, and this takes you to a page that shows you how to use the fonts from the issue in your own programs.

All in all, not a bad first issue, and I can remember loading it for the first time and being amazed that my Spectrum is capable of doing this kind of thing.

Even today, it's a good source of routines to use yourself, and the reviews and programs takes you back to 1983.

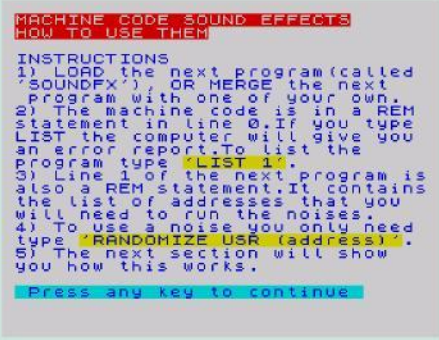
Next we get another game, Chessfire. Chess, but with the option of shooting the other pieces. Not being a chess fan, I can't comment on its engine. It's a manual game and you cannot play the computer.





The next section is a utility to produce machine code sound effects, or rather gives you a selection to use in your own games.

After you have pressed various keys to hear the effect, you can break into the listing and copy various REM statements to use for yourself.



Next we get an animation demonstration. A bird flaps across the screen, and then you get a chance use this, again in your own programs, but have to rip the routines from the main listing. The frames are taken from a SCREEN, so it is easy to make your own.

This forms part of a competition, asking readers to submit their own animations.



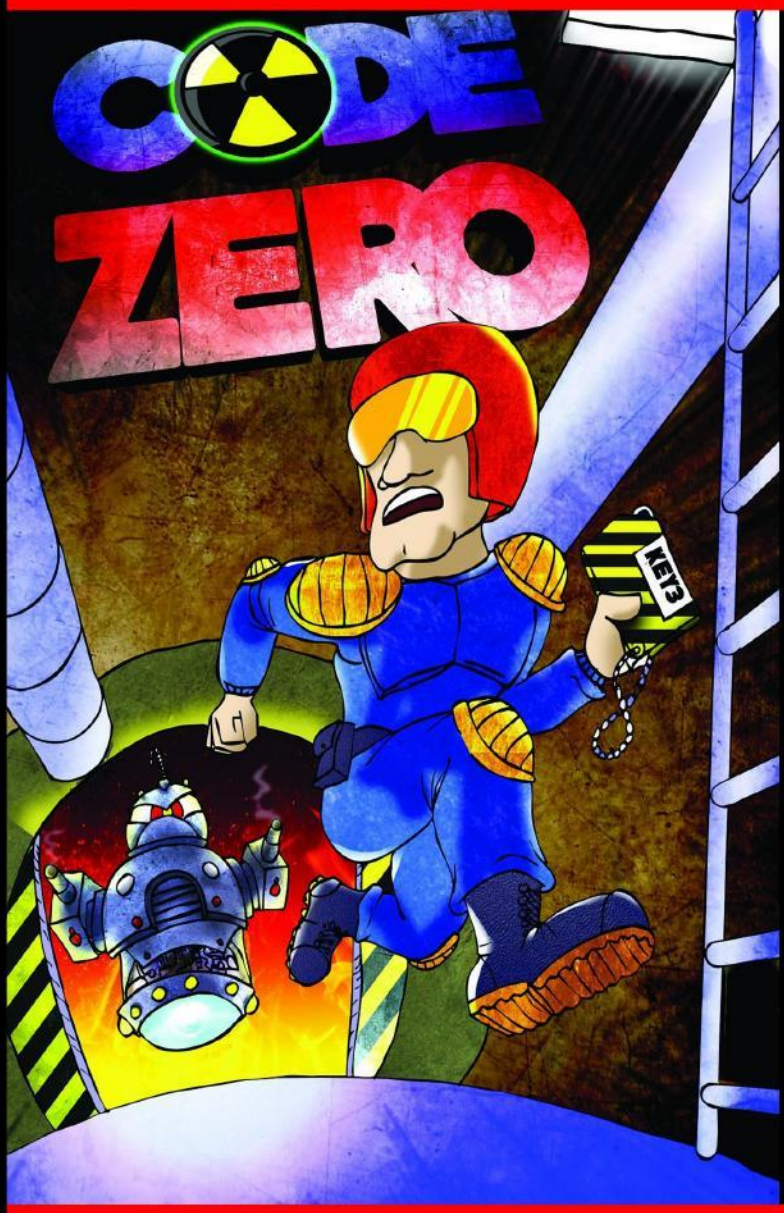
ABOUT 16/48

16/48 was a tape based magazine that began publication in November 1983. It wasn't the first magazine of this type for the Spectrum, that was Spectrum Computing.

16/48 came attached (with glue) to a large A4 sized card with a paper magazine like style on the front, and a list of contents on the back.

The magazine ran from November 1983 to June 1985.

RELEASES ON REAL MEDIA



CODE ZERO

It's about to blow.
Can you be a hero?

Infiltrate the base, watch out for rogue robots, swerve past dripping acid...

Locate the computer room and do your thing...

Action adventure game for the Spectrum 128*



*works on 48k models with reduced sound

ALSO AVAILABLE

Games from Paul Jenkinson



Toofy In Fanland+

Help Toofy get his nuts back in the weird world of fans.



Space Disposal

Cleaning up the universe, one planet at a time.



Deep Core Raider+

Navigate the planets, grab the loot and get out alive.

Bounty

Test yourself with this sc-fi text adventure.

Baldy ZX

Unique plaform game across 20 levels.

www.thespectrumshow.co.uk